(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 December 2000 (28.12.2000)

PCT

(10) International Publication Number
WO 00/78374 A1

(51) International Patent Classification[7]: A61M 1/03, A61B 5/021, G06F 17/60, 15/42, 159/00

(21) International Application Number: PCT/US00/16973

(22) International Filing Date: 21 June 2000 (21.06.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/139,816 21 June 1999 (21.06.1999) US

(71) Applicant: **ELLORA SOFTWARE, INC.** [US/US]; 7 Bean Street, Devens, MA 01432 (US).

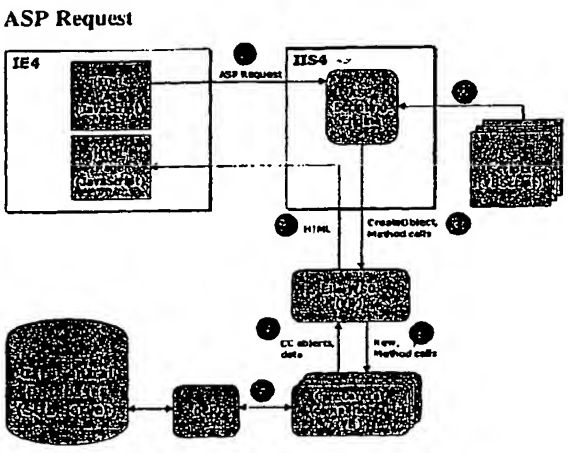
(72) Inventors: **ABEYTA, Rebecca**; Ellora Software, Inc., 7 Bean Street, Devens, MA 01432 (US). **CACCIAPOUTI, Mark**; Ellora Software, Inc., 7 Bean Street, Devens, MA 01432 (US). **DOBES, Katerina**; ** (US). **DUDEK, David**; Ellora Software, Inc., 7 Bean Street, Devens, MA 01432 (US). **ESIELIONIS, John**; Ellora Software, Inc., 7 Bean Street, Devens, MA 01432 (US). **HODGEMAN, Lawrence**; Ellora Software, Inc., 7 Bean Street, Devens, MA 01432 (US). **JENSEN, Keith**; Ellora Software, Inc., 7 Bean Street, Devens, MA 01432 (US). **KATZENBERG, Barbara**; Ellora Software, Inc., 7 Bean Street, Devens, MA 01432 (US). **KIERSTEAD, Leonard**; Ellora Software, Inc., 7 Bean Street, Devens, MA 01432 (US). **KLOKMAN, Lynn**; Ellora Software, Inc., 7 Bean Street, Devens, MA 01432 (US). **LELIEVRE, Nancy**; Ellora Software, Inc., 7 Bean Street, Devens, MA 01432 (US). **MARKSTEINER, Regina**; Ellora Software, Inc., 7 Bean Street, Devens, MA 01432 (US). **MCDERMOTT, John**; Ellora Software, Inc., 7 Bean Street, Devens, MA 01432 (US). **OLEJARZ, Charles**; Ellora Software, Inc., 7 Bean Street, Devens, MA 01432 (US). **PICKARD, Fred**; ** (US). **YOST, Gregg**; Ellora Software, Inc., 7 Bean Street, Devens, MA 01432 (US).

(74) Agents: **KURTZ, Richard** et al.; 11th Floor, 1750 Tysons Boulevard, Tysons Corner, VA 22102 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AT (utility model), AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, CZ (utility model), DE, DE (utility model), DK, DK (utility model), DM, DZ, EE, EE (utility model), ES, FI, FI (utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KR (utility model), KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG,

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR INTERNET-BASED ACTIVITY MANAGEMENT



(57) Abstract: Described are software applications which operate in combination with hardware and use the Internet as a transport to manage medical treatment services. Tasks involved in treating a patient are coordinated by assigning a care treatment plan to each patient. Patient information is first collected from patients, care providers, and other sources. The software then permits a cooperating group to select and create a care treatment plan. By selecting an appropriate care plan template and selecting appropriate care options within that template (7), a care plan specific to the patient's situation can be created which sets forth the care options that typically need to be performed (4). Specific care options and tasks for further treatment can be dependent on the results of prior tests or examinations, or additional information collected from various sources. The software application includes a template instantiation (3) and transaction engine which is able to simultaneously manage the activity templates for thousands of activity instances.

WO 00/78374 A1

MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (utility model), SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND APPARATUS FOR INTERNET-BASED ACTIVITY MANAGEMENT

This application claims the benefit of U.S. Provisional Patent Application No. 60/139,816 filed June 21, 1999. The entire disclosure of that application, including the appendices thereto, is incorporated by reference herein.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates in general to software and hardware for management of activities, and in particular to novel systems and methods for managing activities which use the internet as a transport and which facilitate access by a broad user base.

2. Related Art

Activities that involve a number of people, not all co-located, repeatedly cooperating to achieve a common goal are notoriously hard to manage. One such activity is the provision of evidence-based medical treatment to patients. The provision of evidence-based medical treatment involves a number of care providers and patients, not all co-located, who repeatedly cooperate to operationalize, always under slightly different circumstances, a set of treatment guidelines appropriate to each patient.

Management of the provision of evidence-based medical treatment for chronically ill patients has, heretofore, generally been done manually. Such care typically spans a long period of time, and requires the involvement of a number of care providers: physicians, nurse practitioners, specialists, test laboratories, pharmacies,

etc. Once a diagnosis for an illness has been made, a physician generally prescribes treatment by marking on a paper chart, or writing a note to an assistant. There are several steps involved in following up with the treatment, such as ordering lab tests, reviewing results, ordering additional tests, prescribing medications, checking the results of medications, scheduling additional visits, providing educational material. These tasks can multiply significantly when a number of patients with different illnesses are involved.

When prior art methods are used, substantial time can be wasted in keeping track of what needs to be done for whom. There is no consistent method of communicating between the various care providers. Information can be misinterpreted, a note could be misplaced or lost, resulting in inappropriate care or no care, which could be costly.

In prior art systems for care management, the treatment plan for a patient usually consists of a paper with a high level prescription or a list of common tasks or responses to be carried out. The information is typically vague and not specific to a patient's specific, changing medical conditions. Prior art systems lack an automated process for selecting the right set of care options and associated tasks, and assigning them to a patient based on the patient's specific condition. As a result, it is not possible to effectively and efficiently create a treatment plan specific to each individual patient, and share it with all the care providers. The care treatment plan documentation that exists is generally in the form of a flat document listing all possible treatment variations. It is difficult to create, act upon and manage a patient treatment plan from such a flat document in each instance.

Evaluation of next steps has been based upon additional paper-based information, which may not be up-to-date or accurate. To properly prescribe the next treatment option, information from several sources, including the patient, needs to be

collected and evaluated. Previously, such information was collected from patients and other providers via telephone, personal visits, or paper. This collection process is time-consuming, and inconsistent, which delays the treatment.

OBJECTS AND SUMMARY OF THE INVENTION

It is therefore an object of the invention to provide an improved system and method for activity management.

It is a further object of the invention to provide an improved system for managing evidence-based medical treatment services.

In accordance with the invention in its preferred embodiment, a set of software applications are provided which operate in combination with hardware and use the internet as a transport to manage medical treatment services. Tasks involved in treating a patient are coordinated by assigning a care treatment plan to each patient. Patient information is first collected from patients, care providers, and other sources. The software then permits a cooperating group to select and create a care treatment plan from a hierarchical treatment plan template library consisting of a large collection of care plan templates, each specific to a particular illness type or subtype. Each care plan template consists of a different combination of care options. Each care option consists of a different combination of tasks (e.g., Office Visit, EKG, blood test, medication, etc.). By selecting an appropriate care plan template and selecting appropriate care options within that template, a care plan (i.e. a collection of tasks) specific to the patient's situation can be created which sets forth the care options that typically need to be performed, when those tasks need to be performed, and by whom, to achieve each goal variant.

The software in its preferred embodiment provides a management system which permits individuals within the cooperating group to further specify, in very little time, the activity template and activity instances therein so that they are aligned with a goal variant. While an activity is being performed, tasks whose relevance to the goal variant could not be predetermined can be dynamically evaluated to determine whether or not they need to be performed. Specific care options and tasks for further treatment can be dependent on the results of prior tests or examinations, or additional information collected from various sources. The software application includes a template instantiation and transaction engine which is able to simultaneously manage the activity templates for thousands of activity instances.

Access to the system by users is preferably provided via a web browser. In this respect, members of the cooperating group have access to the software application no matter where they are – without any special infrastructure requirements. Access to the system by each member of the cooperating group can be limited to those pieces of information (e.g., tasks) that are relevant to that member.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, features, and advantages of the invention will be apparent from the following more particular description of preferred embodiments as illustrated in the accompanying drawings, in which reference characters refer to the same parts throughout the various views. The drawings are not necessarily to scale, emphasis instead being placed upon illustrating principles of the invention.

FIG. 1 is a diagram illustrating a hierarchical task list in accordance with a preferred embodiment of the invention.

FIG. 2 shows a diagram illustrating the automatic task scheduling feature of the invention in accordance with a preferred embodiment.

FIG. 3 shows a diagram illustrating the preferred system and method of the invention for collecting relevant clinical information from geographically distributed multiple participants in an asynchronous manner, using conditional questionnaires over the internet.

FIG. 4 shows a block diagram illustrating the architecture of the system through which the invention is delivered according to a preferred embodiment.

FIG. 5 shows a block diagram illustrating the flow of data among the components of the system of the invention.

FIG. 6 shows a block diagram illustrating the flow of data among the components of the system of the invention in connection with an XML request.

FIG. 7 shows a diagram illustrating two ODC objects and their relationships.

FIG. 8 shows a diagram illustrating the ODC driver's relation to ODC Elements.

FIGS. 9-13 show screenshots illustrating the user interface in the preferred embodiment of the "CareCentral" software of the invention.

FIGS. 14-19 show screenshots illustrating the user interface in the preferred embodiment of the "Customizer" software of the invention.

DETAILED DESCRIPTION

The invention in its preferred embodiment provides a system for managing evidence-based medical treatment services. For example, a breast cancer management system can be provided using the invention. In the context of such a system, the invention can be used to manage the care of patients who need to undergo a number of procedures to determine whether or not they have breast cancer and to manage the treatment of patients who have been diagnosed with breast cancer. In both cases, the invention preferably provides functionality:

- to collect from everyone involved in the care of the patient, including the patient himself, just the information that care providers need in order to determine the patient's state,
- to enable the activities and tasks that need to be performed to be continually refined in order that the patient can continue to be treated appropriately,
- to enable everyone who is responsible for performing tasks, no matter where they are, to be reminded when each task needs to be performed,
- to enable everyone who is responsible for performing tasks, no matter where they are, to get substantial automated assistance in performing the tasks,
- to enable the efficacy of each of the actions and tasks - for a particular population of patients -- to be evaluated,
- to enable the templates that define what actions, tasks, and question sets are appropriate in what circumstances to be modified by the user organization based on the results of the efficacy evaluations.

The invention preferably uses the internet, or a secure, virtual private network within the internet, as a data transport. Alternative embodiments may use other types of private wide area networks, such as an intranet or an extranet, without departing from the spirit and scope of the invention.

The invention in its preferred embodiment comprises two software artifacts, which are respectively referred to herein as the Customizer and CareCentral. The Customizer enables the members of a cooperating group to create activity templates; these templates identify possible activity options and, for each option, specify what tasks need to be performed and what information needs to be collected in order for the activity (and its management) to be accomplished.

CareCentral, which is accessed via a user's web browser, enables a member of the cooperating group involved in a particular activity instance to further specify, in a few seconds, which activity options need to be performed to achieve the goal implied by that instance. CareCentral monitors the situation and, based on all of the information it has been provided with to date, dynamically evaluates whether or not certain conditional actions specific to this activity instance are required. CareCentral's template instantiation and transaction engine is able to simultaneously manage the activity templates for thousands of activity instances.

FIG. 1 is a diagram illustrating a hierarchical task list in accordance with a preferred embodiment of the invention. The Customizer tool is used to generate a hierarchical task list comprising the tasks always implied for any patient with a particular problem, and also the tasks that may be implied depending on the patient's particular situation. Since a "flat" collection of tasks can be difficult to manage, an activity template is created which consists of a hierarchy of tasks that are distributed across multiple Care Plan Templates (CT) related to different clinical conditions. Each

care template consists of multiple Care Options (CO). Each care option consists of multiple Tasks (T).

Each care option and task can have a "YES", a "NO" or a "MAYBE" state of specification. Based on the clinical condition, certain care options and certain tasks will have an automatic "YES" state specified. Based on the patient's observed condition, relevant care options and tasks with "MAYBE" conditions can be switched to "YES" or "NO". This method allows only the tasks relevant to a patient's particular state to be identified for performance.

FIG. 2 shows a diagram illustrating the automatic task scheduling software feature of the invention in accordance with a preferred embodiment. This feature provides for automatically scheduling tasks that are not normally implied, i.e., set to "YES." Such tasks are automatically added to a task hierarchy instantiation by applying the relevant set of rules to information collected about the patient's particular situation.

To accomplish automatic task scheduling, a set of "rules" are first specified within each care plan and care option in a hierarchy to examine relevant information identified as information objects, I1, I2....In, collected from multiple sources: interactive questionnaires, external databases etc. When conditions specified for a rule are satisfied, a care option set to "YES" or a task set to "YES" is automatically added to the task hierarchy instantiation. This method allows only the tasks relevant to a patient's particular state to be automatically identified for performance.

FIG. 3 shows a diagram illustrating the preferred system and method of the invention for collecting relevant clinical information from geographically distributed multiple participants in an asynchronous manner, using conditional questionnaires over the internet. The population of personnel involved in the medical treatment of a

patient is usually geographically dispersed. This system and method provides a means for asking a group of geographically distributed participants a set of questions which are relevant to a particular participant's particular situation, and gathering the resulting information to enable the automatic scheduling of hierarchical tasks.

The internet is preferably used to present to each participant (including the patient) a questionnaire consisting of multiple questions related to a patient's particular situation. The questions are conditioned such that only the relevant questions are displayed, based on the answers received to previous questions. Questions not relevant are hidden from view.

This mechanism allows only the relevant information to be collected from a wide variety of sources. The information can be collected in an asynchronous fashion, whenever it is needed, or whenever the participant can provide it, in a manner whereby timing is not critical. Participants may provide information from wherever they are - from geographically distributed locations. The information collected is then used by the automatic task scheduling software discussed above. The rules associated with automatic task scheduling are applied to the information to automatically schedule relevant tasks for a particular patient.

SYSTEM ARCHITECTURE OF THE PREFERRED EMBODIMENT

The system architecture in accordance with a preferred embodiment of the invention is described below. It will be apparent to those skilled in the art that other architectures are possible without departing from the spirit and scope of the invention.

Reference is now made to FIG. 4, which shows a block diagram illustrating the architecture of system through which the invention is delivered according to a preferred embodiment. The architecture includes a client, presentation services, application services, data services, and databases. Each of these elements is described below.

The client preferably comprises a DOM-compliant web browser or Internet Explorer 4, but may comprise another suitable web browser. The presentation services portion of the architecture preferably includes a web server supporting Active Server Pages along with a web presentation layer. Active Server Pages are used to allow efficient dynamic generation of web pages. Because both the content and the layout of web pages are data driven in CareCentral, this efficient dynamic generation of web pages provides significant advantages. For example, by using Active Server Pages, data from a database (described in detail below) can be assembled and the best presentation form can be selected on-the-fly. The web server preferably delivers data to the client in HTTP-, HTML-, and XML-compliant formats.

The use of Dynamic HTML allows web pages to change and interact with the user without requiring the user to navigate to a new page. This makes the CareCentral application continuously usable as part of a user's ordinary work day. XML is used for its capabilities for describing richly-structured data such that it can be processed by off-the-shelf software. JAVA is used on both the client and the server. On the client side, invisible JAVA applets manage web page operation and presentation by manipulating Dynamic HTML objects. On the server, JAVA is used for generating web pages. The Web Presentation Layer dynamically constructs web pages based on information stored in the business objects and databases. The web browser requests a web page, specifying what information is needed in the web page's URL. The web presentation layer queries the business objects for the required information and formats

it as appropriate HTML or XML pages that are then sent back to the user's web browser.

The application services layer preferably includes XML object services, business objects, and a reporting/integration data extractor client. The business objects encapsulate all of the basic knowledge about how the various entities used in the system of the invention operate (example entities include Patients, Users, and Care Plans). The business objects present a high-level view onto these entities, hiding the details of their representation. This ensures database consistency and simplifies the programming of operations on these entities.

The reporting/integration data extractor client allows users to identify sets of interrelated data that they want to extract from the system's databases, for example because they want to produce a report or export the information to an external system. The data extractor client accepts the description of the data the user wants and passes it along to the data extraction server (described below). The server processes the query and returns the results to the client, which then formats the results into the form the user specified.

The XML object services are a layer built on top of both the business objects and the data extractor client. The XML object services accept requests for business objects operations and data extraction operations expressed as XML documents. The XML services parse the XML request document, call on the business objects and the reporting extractor to perform the requested operations, and then format the operations' results as XML response and status documents that are returned to the requestor.

The data services layer preferably includes an object/relational server and a reporting/integration data extractor server. The reporting/integration data extractor

server processes requests passed to it from the data extractor client component (described above). It queries the system's databases and business objects for the required information and passes the results back to the data extractor client for further formatting and presentation to the requester. The object/relational server mediates between the business objects and the relational databases that actually store the data. The business objects make object-oriented requests for data and data updates. The object/relational server maps these object-oriented requests into relational database queries, executes the queries, and builds and returns object structures representing the relational query results. Data interchange between the data services layer and the application services layer is accomplished using CORBA distributed object services. The use of these services allows CareCentral software components to be partitioned across multiple computers. This further enables efficient client-server operation and improves web scalability.

Data is stored in databases which preferably include a Microsoft SQL Server and an Oracle database. Access to the data by other components of the system is accomplished using the ODBC/JDBC protocols, which have broad multi-vendor database support.

FIG. 5 shows a block diagram illustrating the flow of data among the components of the system of the invention in connection with an ASP Request. At step 1, an event handler on an HTML page (e.g., a JavaScript function) causes an ASP request. In response, at step 2, the ASP scripting engine in the web server software (i.e., IIS4) finds the requested page and executes its server-side script. Then, at step 3, the server-side script code creates an instance of an EllorWSO object and calls its methods. At step 4, EllorWSO methods create CareCentral business objects and call their methods. At step 5, CareCentral business object methods interact with the CareCentral databases via OCP. At step 6, CareCentral business object methods return CareCentral objects

and other data to EllorWSO. Finally, at step 7, EllorWSO transforms those objects and data into a new HTML page, which is sent directly to the client browser via the web server software.

FIG. 6 shows a block diagram illustrating the flow of data among the components of the system of the invention in connection with an XML request. At step 1, an event handler on an HTML page (e.g., a JavaScript function) calls methods of a Java class on the client. At step 2, one of the Java methods on the client called in step 1 expects XML to be returned. This method causes an ASP request. This method will wait for XML to be returned and then parse the XML and create appropriate data structures. At step 3, the ASP scripting engine in the web server software (e.g., IIS4) finds the requested page and executes its server-side script. The server-side script code, at step 4, creates an instance of an ElloraWSO object and calls its methods. At step 5, ElloraWSO methods create CareCentral business objects and call their methods. At step 6, CareCentral business object methods interact with the CareCentral databases via OCP. At step 7, CareCentral business object methods return CareCentral objects and other data to ElloraWSO. At step 8, ElloraWSO creates Java objects on the server and calls their methods to transform CareCentral objects and data into XML. At step 9, XML is sent back to ElloraWSO as a result of Java method calls on the server. At step 10, ElloraWSO passes the XML back to the requesting Java method on the client via the web server software. The requesting Java method on the client parses the XML. At step 11, the JavaScript function that was initiated in step 1 calls additional Java methods on the client for displaying the parsed XML data returned in step 10. Finally, at step 12, Java methods on the client transform the parsed XML into HTML, and insert it into the same HTML page that made the initial request in step 1.

OPEN CARE PLATFORM

The system in accordance with the invention preferably includes middleware which provides a uniform method of access by client applications to different data sources. The middleware is referred to herein as the Open Care Platform, or OCP. The OCP middleware provides an object interface to persistent objects contained in a database, and simplifies defining persistent objects, creating databases that contain those objects, and using those objects. OCP *instances* each serve a different data source, and OCP clients can invoke multiple OCP instances to access different data sources. OCP is designed to provide access to different *types* of data. The OCP preferably supports relational databases via ODBC or other database standard.

The OCP is based on distributed object technology and is preferably designed to support multiple distributed object models. The OCP can be designed to support the Object Management Group's Common Object Request Broker Architecture (CORBA) standard. Both SmallTalk and Visual Basic (via OLE) clients have used CORBA-based access to OCP. Microsoft's Distributed Common Object Model (DCOM) for client access via Active X may also be supported.

At the highest level, developing an OCP instance for use by client applications involves:

1. Developing or selecting a data source (initially Oracle and SQL Server data are supported);
2. Developing a client-side OCP Application Programming Interface (API) to the specific data source; and
3. Developing a data-source-specific, server-side OCP interface to the OCP core.

Note that the application programmer preferably deals only with the client API.

The OCP *instance*-development process is designed so that each of steps 1-3 above may be automated by the use of code-generation tools. For example, Oracle's Designer 2000 data modeling tool (with existing extensions) can be used to do all of the following:

- design the database
- generate the Data Definition Language (DDL) statements which
- create the database
- generate the Object Relational Map (ORM) file which OCP tools use to create both the server-side OCP interface, and the client-side OCP API.

Additional modeling tools, such as Rational Rose, may be supported.

An OCP core is preferably provided and acts as the kernel of OCP-- it does not change with the data source or the distributed object technology. Both client and server OCP interfaces are data source specific.

OPENDATACONTROL (ODC) COMPONENT

Since a significant purpose of components and applications in accordance with the invention is to provide a user interface to data stored in potentially a variety of data sources, it is advantageous to have a relatively simple method of linking a visual element (e.g. a Visual Basic control such as a TextBox, ListBox, OptionButton, etc.) to a data element. Some existing Visual Basic controls, known as *bound controls*, allow this linking through the VB Data control, but the linking must be to a specific field within a specific database. Since Ellora components and applications may not necessarily know the exact database name and data model of the data they will be displaying, it will not be possible to use bound controls and the Data control.

Therefore, a significant goal of the OpenDataControl (ODC) is to link together visual elements and data elements, allowing the data elements to come from a variety of data sources. In addition, ODC will also go beyond simple linking of visual elements to data elements to support a number of other requirements, as outlined in the next section. Due to the nature of ODC as the provider of a linkage between visual and data elements, it will not be responsible for any specific user interface designs or database models.

As stated above, the main goal of ODC is to link together visual elements and data elements. In addition, ODC will support a number of other requirements that apply generically to either visual elements or data elements or both at a low level. The other requirements are as follows.

1. Support of default values

It is often the case that certain data elements in an application can have default values, which can save the user time when using the application. Default values may be simple strings or numbers that can be expressed absolutely, or they may be calculations that can depend upon other data elements. Therefore it would be beneficial if ODC supported assigning of default values to data elements.

2. Support of red flags

Certain applications may want to know when the value entered for a certain data element is out of a specified range, also known as a *red flag*. It would also be useful to have a method for visually indicating this situation. Therefore it would be beneficial if ODC supported red flags.

3. Support of validation

Before a data element can be committed to a data source, there is almost always some level of validation that needs to be done on that data element. This can include simple validation such as validation of data type, dates and times, and it can also include more complex validation that requires an arbitrary function to be run to perform the validation. Therefore it would be advantageous for ODC to support validation.

4. Default event routines

Certain VB event routines may want to call ODC methods in response to user actions. It is also likely that the code for certain events for certain VB controls (e.g. the LostFocus event of a TextBox control) will be the same for all VB controls of that type. It would therefore be advantageous if ODC supported default routines for certain events based upon the VB control type so that most VB event routines would simply have to make one method call to ODC.

Objects Provided and Their Interfaces

ODC comprises three objects, implemented as Visual Basic classes. The two objects that are of primary concern to the user of ODC are the *Element* object and the *Elements* collection object. These two objects and their relationships are depicted FIG. 7.

An Elements object is a collection object comprising one or more Element objects. Conceptually, each Element object links one or more visual elements (VB controls) with one data element (attribute of some object). Instantiated Element objects are first programmed with the proper visual element(s), data element and behavior properties, and then methods of Elements collection objects and/or Element objects are called to perform useful work such as retrieving data from data sources, displaying data in controls and storing data to data sources. Each Element object must belong to an

Elements collection object. In other words, the Element class' Instancing property will be set to NotCreatable, and an Element will have to be created via the "Add" method of an Elements object.

Since ODC is a low-level component that basically just understands linking of visual and data elements, there will be no support in ODC for querying a data source, and ODC will not understand the concept of a database session. Therefore the user of ODC will have to perform those functions that fall into these categories outside of ODC. Once a query is performed, though, certain ODC properties can be set and certain methods can be called that will operate on the results of the query.

The third object that ODC contains is an *ODC driver* object. The ODC driver separates the logic contained in the Elements and Element objects to tie together visual and data elements, from the logic needed to access data via OCP (or potentially another data access mechanism). In other words, the ODC driver encapsulates all OCP calls made by ODC so that neither the Elements or Element objects have to make any calls directly to OCP. ODC supplies two generic ODC drivers, ODCDCPDriver and ODCOCPDriver. The ODC driver's relation to ODC Elements is depicted in FIG. 8.

The flexibility inherent in using a driver allows ODC Elements to potentially access any data source in the same manner without knowing the details of how to access that data source. This can be accomplished by replacing a generic ODC driver with another ODC driver that implements the details of accessing another data source. As long as the new ODC driver has the same interface as the published generic ODC Driver interface, described in Section 3.3, ODC can use it. (Note that the same interface does not necessarily imply identical, but at least compatible). The ODC driver to be used by ODC Elements is programmed via the ODCDriver property.

Element Object

The Element object is the primary object around which the design of ODC revolves. The primary goal of an Element object is to tie together one or more VB controls and one piece of data. In general, an Element is instantiated for each control on a VB form, via the Elements collection object (see next section) and then its properties are set. Once the Element has been instantiated and initialized, its methods are called to load its value into its control, get and put its value, check if its value has changed, and perform other processing.

An important concept surrounding the Element object is that of the *DataSourceValue* vs. the *CurrentValue*. Conceptually, each Element object stores two values per data element, the DataSourceValue, which is the value that is stored in the data source for that data element, and the CurrentValue, which is the value that the user has entered or changed for that data element, usually via the Element's VB control. (Note that they are the same when the VB control is first loaded from the data source.) The DataSourceValue is thus obtained directly from the appropriate data source, whereas the CurrentValue is usually the value of the VB control. Note that there may be situations where the CurrentValue cannot be stored in the VB control, such as when the Element does not contain a VB control, or when the VB control is not dedicated.

Another important concept to understand is that some properties will need a query to be performed to be set, and others will not. The process of setting the values of properties that do not need a query to be set is referred to as *initialization*, and the process of setting the values of properties that do need a query to be set is referred to as *binding*. The main reason for making this distinction is to point out that, typically, initialized properties will only be set once during the life of the application (or instance of a specific VB Form), for example in the Form_Load event, whereas bound properties may have to be set at various points in the application, for example when a

new patient is selected, when a new view is selected, or when a new data entity is created. This implies that a given set of properties of an Element should only be initialized once, and another set can be bound many times to allow display of the same set of data for different entities.

Elements Object

The Elements object is a collection object that contains a collection of Element objects. The only means of instantiating an Element object is via the **Add** method of the Elements object.

The main purpose of the Elements collection object is to provide properties and methods that allow manipulation of all Element objects in the collection, thus providing convenient processing for the user of ODC. In general, the Elements object property or method will set or return the property or method with the same name in all Element objects in the collection. This implies that many properties and methods in this category do not have to be called at the Elements object level, but can always instead be called for each individual Element in the collection.

There are two major exceptions to the above category of properties and methods. The first is those methods that directly manipulate the collection itself. These include the Count property, and the Add, Remove and Item methods, which are directly analogous to those properties and methods in the general VB Collection object. (Note that the Elements Add method does add some value to the general VB Collection method, such as instantiating an Element object.)

The second major exception to the above category is that the Elements object is the only place where methods associated with processing a grid control are located. The reasoning behind this is that a grid control contains multiple Element objects, each used as a template for a grid column, and it seems logical for those methods that affect

or use more than one Element object to belong in the Elements collection object. These methods include LoadGrid, GetGridCell, StoreGridCell and the UnboundReadData and UnboundWriteData standard event handlers.

USER INTERFACE OF THE PREFERRED EMBODIMENTS

FIGS. 9-13 show screenshots illustrating the user interface in the preferred embodiment of the "CareCentral" software of the invention. FIG. 9 shows aspects of the user interface relevant to planning a patient's care. In particular, FIG. 9 illustrates the interface to functions for viewing and modifying individual care plans, adding to a patient's overall plan, capturing data associated with the care provided, choosing from a set of care templates customized for a user's care practices, and customizing the care templates for an individual patient.

FIG. 10 shows aspects of the user interface relevant to recording a patient's progress. In particular, FIG. 10 shows illustrates the interface to functions for recording data relevant to the care provided, updating patient care plans, and providing notes of patient progress.

FIG. 11 shows aspects of the user interface relevant to coordinating the work required to deliver patient care. In particular, FIG. 11 shows illustrates the interface to functions for tracking work being done, recording results of work which has been completed, and filtering a task list to most efficiently view work being done, and viewing work which has been completed, is overdue, or is in progress.

FIG. 12 shows aspects of the user interface relevant to analyzing outcomes and costs of care delivered. In particular, FIG. 12 illustrates the design and generation of various reports on clinical data collected.

FIG. 13 shows a menu and icon bar useful for navigating the CareCentral component of the invention.

FIGS. 14-19 show screenshots illustrating the user interface in the preferred embodiment of the "Customizer" software of the invention. FIGS. 14 and 15 show aspects of the user interface relevant to care template customization. In particular, these figures show the interface for modification of template definitions.

FIGS. 16 and 17 show aspects of the user interface relevant to care option customization. In particular, these figures illustrate the interface for modifying care option definitions.

FIG. 18 shows aspects of the user interface relevant to customization of the definition of "associated data," i.e., data associated with care options. FIG. 19 shows aspects of the user interface relevant to defining custom reports.

While the invention has been particularly shown and described with reference to a preferred embodiment thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention. For example, the system may be applied to the scheduling of tasks in fields other than the provision of medical care services, such as the commercial real estate lending field, within the scope of the invention.

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. A method for managing tasks associated with the provision of medical care, comprising the steps of:

providing software which receives patient information identifying at least one patient, said patient information comprising information regarding medical conditions of said at least one patient;

using said software to select for said at least one patient a treatment plan template comprising a task list, said task list comprising a first set of tasks and an indication that they are scheduled for said at least one patient, and further comprising a second set of tasks and an indication that they may be scheduled for said at least one patient depending upon said at least one patient's particular medical conditions;

using said software to schedule tasks by applying a set of rules to said information regarding medical conditions and, when said rules are satisfied, scheduling at least one task in said second set of tasks.

2. A method for managing tasks associated with the provision of medical care, comprising the steps of:

using software to create and store a database of patient information, said database comprising, for each of a plurality of patients, at least one care option having a "yes" or a "maybe" state associated therewith and a plurality of tasks associated with said at least one care option, each task having a "yes" or a "maybe" state associated therewith.

3. A system for managing tasks associated with the provision of medical care, comprising:

a computer system operating in accordance with software to create an activity template comprising a hierarchy of tasks distributed across multiple care plan templates, each care plan template having a plurality of care options associated therewith, each care option having a plurality of tasks associated therewith.

4. The system according to claim 3, wherein each of said plurality of care options has a "yes," "no," or "maybe" state associated therewith.

5. The system according to claim 3, wherein each of said plurality of tasks has a "yes," "no," or "maybe" state associated therewith.

6. The system according to claim 4, wherein said software further comprises functionality for changing care options with a "maybe" state to a "yes" state or a "no" state based upon a patient's observed conditions.

7. The system according to claim 5, wherein said software further comprises functionality for changing tasks with a "maybe" state to a "yes" state or a "no" state based upon a patient's observed conditions.

8. The system according to claim 4, wherein said software further comprises functionality for automatically setting care options to a "yes" state based upon a patient's observed conditions.

9. The system according to claim 5, wherein said software further comprises functionality for automatically setting tasks to a "yes" state based upon a patient's observed conditions.

10. A system for scheduling tasks, comprising:

a computer system operating in accordance with software to provide a questionnaire over a network to geographically distributed participants, said questionnaire including a first question and a second question displayed in response to an answer given by a participant to said first question, said software further comprising functionality for scheduling tasks to be completed based upon a plurality of answers received in response to at least said first and second questions.

11. The system for scheduling tasks in accordance with claim 10, wherein said geographically distributed participants comprise patients and medical care providers.

12. The system for scheduling tasks in accordance with claim 10, wherein said network comprises the internet and wherein said computer system provides said computer system provides said questionnaire via HTML pages.

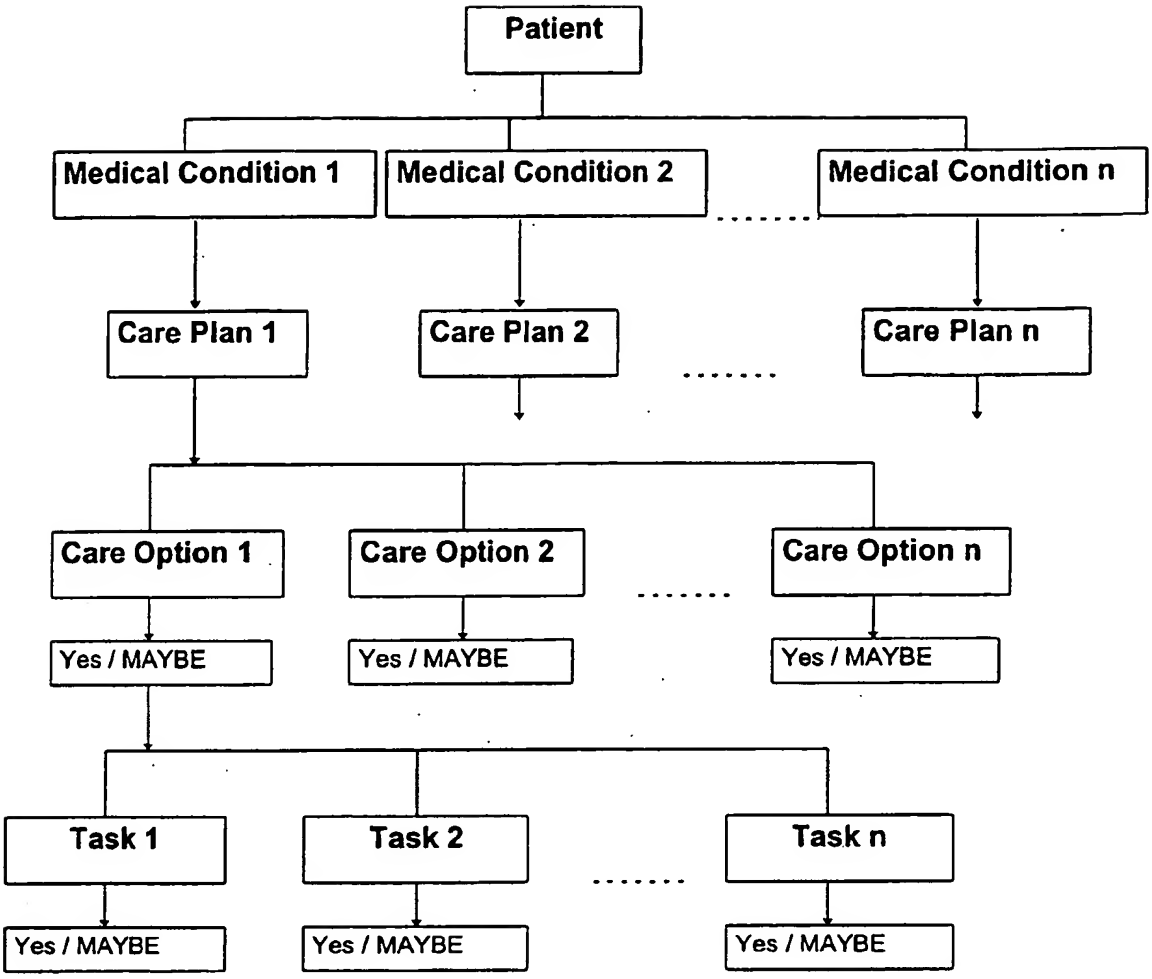

Patient
Medical Condition 1
Medical Condition 2
Medical Condition n
Care Plan 1
Care Plan 2
Care Plan n
Care Option 1
Care Option 2
Care Option n
Yes / MAYBE
Yes / MAYBE
Yes / MAYBE
Task 1
Task 2
Task n
Yes / MAYBE
Yes / MAYBE
Yes / MAYBE

FIG. 1

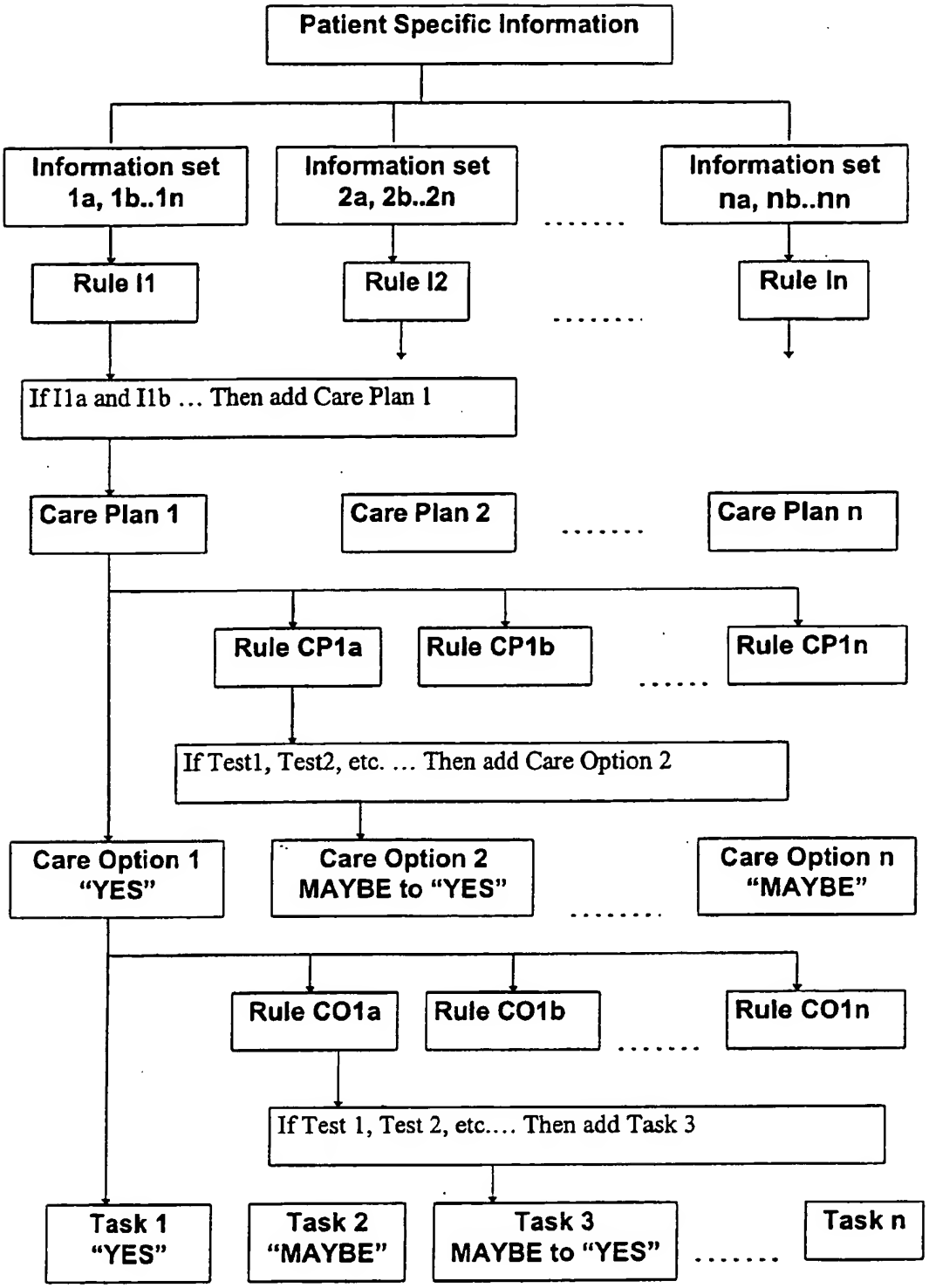

Patient Specific Information
Information set 1a, 1b..1n
Information set 2a, 2b..2n
Information set na, nb..nn
Rule I1
Rule I2
Rule In
If I1a and I1b ... Then add Care Plan 1
Care Plan 1
Care Plan 2
Care Plan n
Rule CP1a
Rule CP1b
Rule CP1n
If Test1, Test2, etc. ... Then add Care Option 2
Care Option 1 "YES"
Care Option 2 MAYBE to "YES"
Care Option n "MAYBE"
Rule CO1a
Rule CO1b
Rule CO1n
If Test 1, Test 2, etc.... Then add Task 3
Task 1 "YES"
Task 2 "MAYBE"
Task 3 MAYBE to "YES"
Task n

FIG. 2

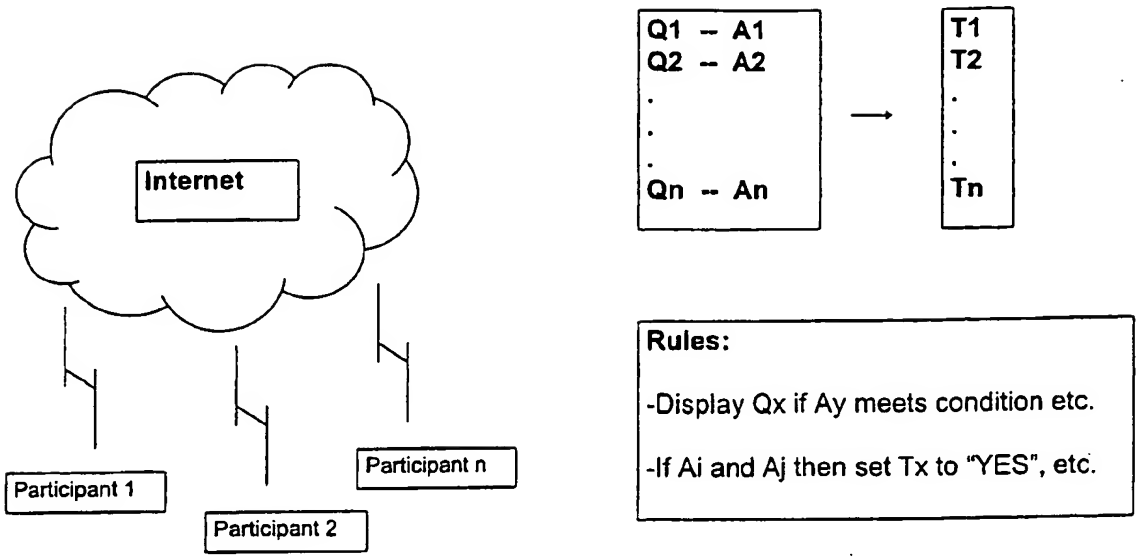

Internet
Participant 1
Participant 2
Participant n
Q1 -- A1
Q2 -- A2
Qn -- An
T1
T2
Tn
Rules:
-Display Qx if Ay meets condition etc.
-If Ai and Aj then set Tx to "YES", etc.

FIG. 3

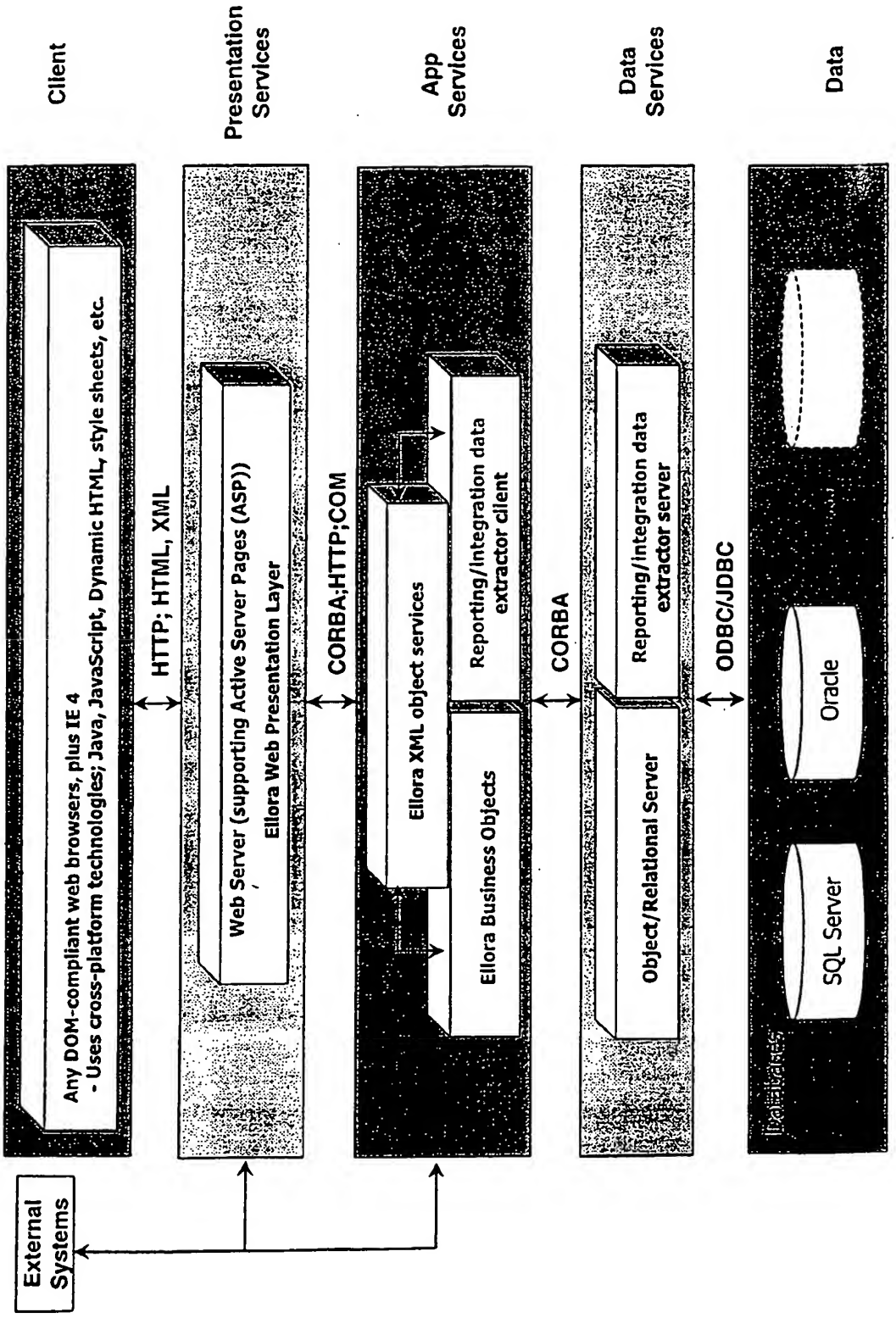

ellora.com Web Architecture
Client
Presentation Services
App Services
Data Services
Data
Any DOM-compliant web browsers, plus IE 4
- Uses cross-platform technologies; Java, JavaScript, Dynamic HTML, style sheets, etc.
HTTP; HTML, XML
Web Server (supporting Active Server Pages (ASP))
Ellora Web Presentation Layer
CORBA;HTTP;COM
Ellora XML object services
Ellora Business Objects
Reporting/Integration data extractor client
CORBA
Object/Relational Server
Reporting/Integration data extractor server
ODBC/JDBC
Oracle
SQL Server
External Systems

FIG. 4

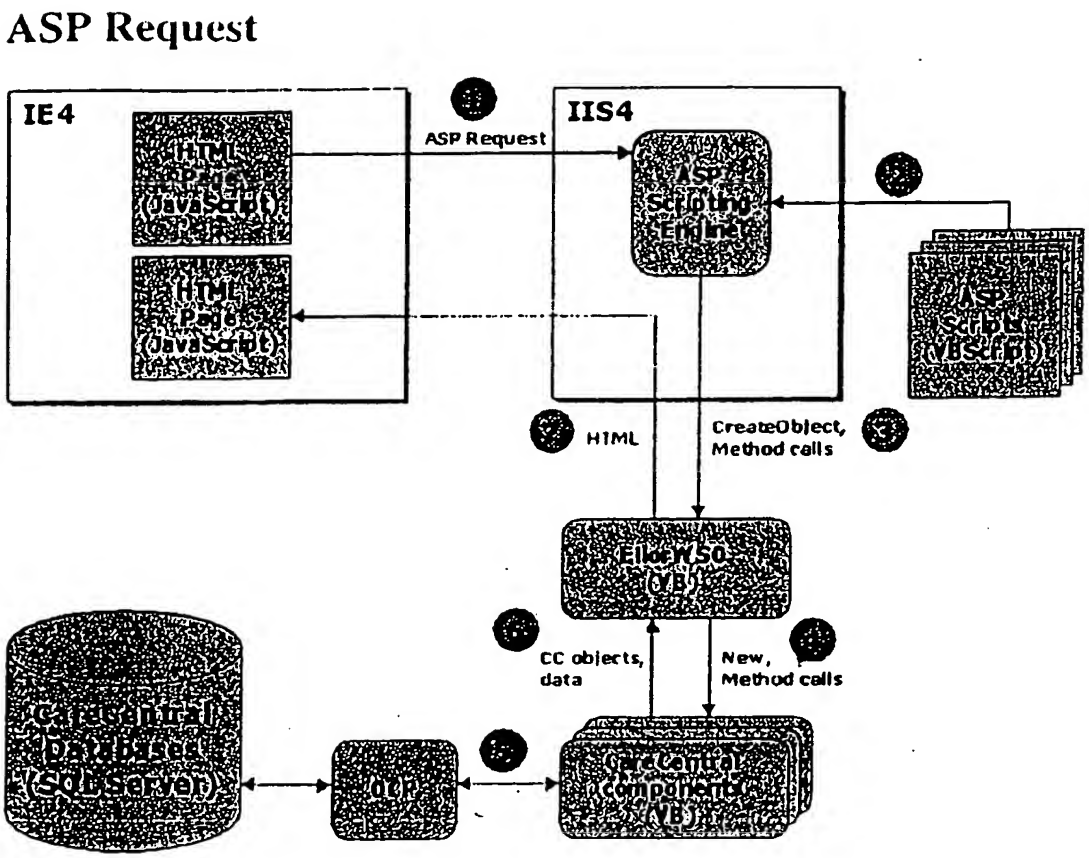

ASP Request
IE4
HTML Page (JavaScript)
HTML Page (JavaScript)
ASP Request
IIS4
ASP Scripting Engine
ASP Script (VBScript)
HTML
CreateObject, Method calls
CC objects, data
New, Method calls
(SQL Server)
OCP
(VB)

FIG. 5

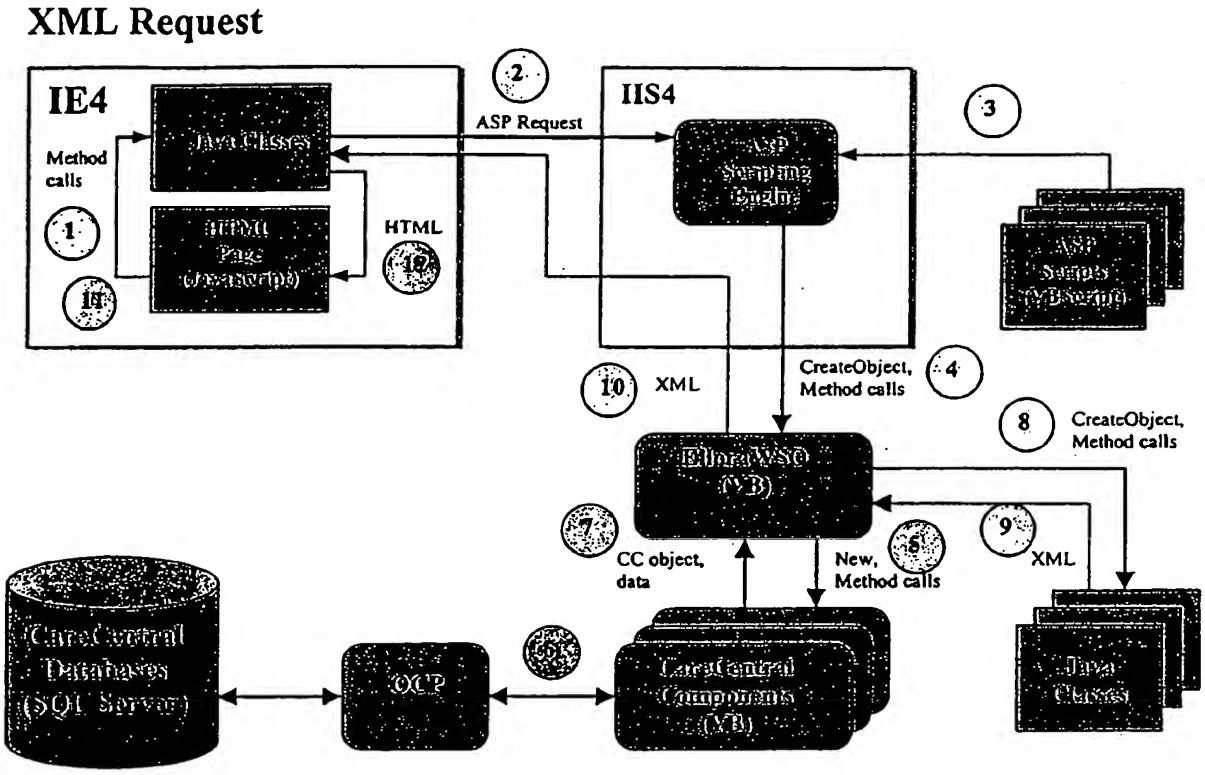

XML Request
IE4
Java Classes
Method calls
1
HTML Page (Javascript)
HTML
12
11
2
ASP Request
IIS4
ASP Scripting Engine
3
ASP Scripts (VBScript)
10
XML
CreateObject, Method calls
4
8
CreateObject, Method calls
ElitraVSO (VB)
7
CC object, data
New, Method calls
5
9
XML
CareCentral Databases (SQL Server)
OCP
6
CareCentral Components (VB)
Java Classes

FIG. 6

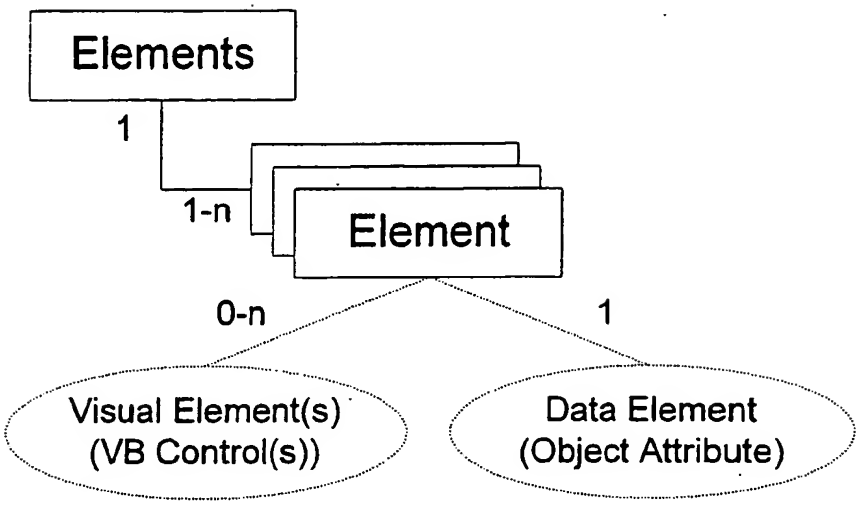

Elements
1
1-n
Element
0-n
1
Visual Element(s)
(VB Control(s))
Data Element
(Object Attribute)

FIG. 7

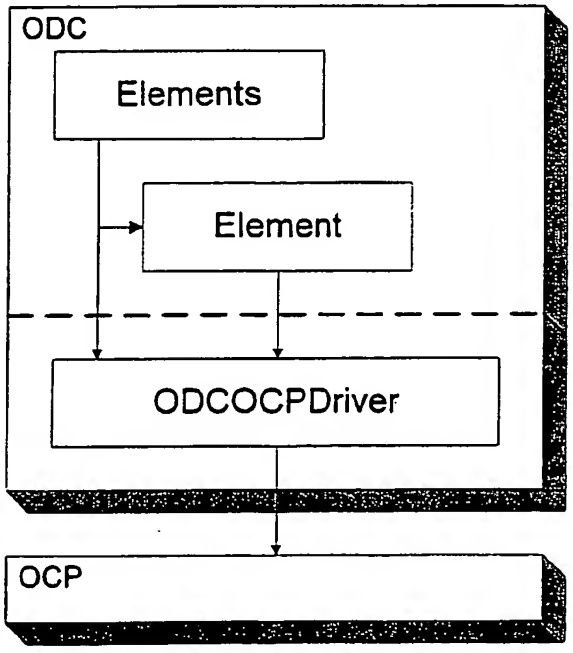

ODC
Elements
Element
ODCOCPDriver
OCP

FIG. 8

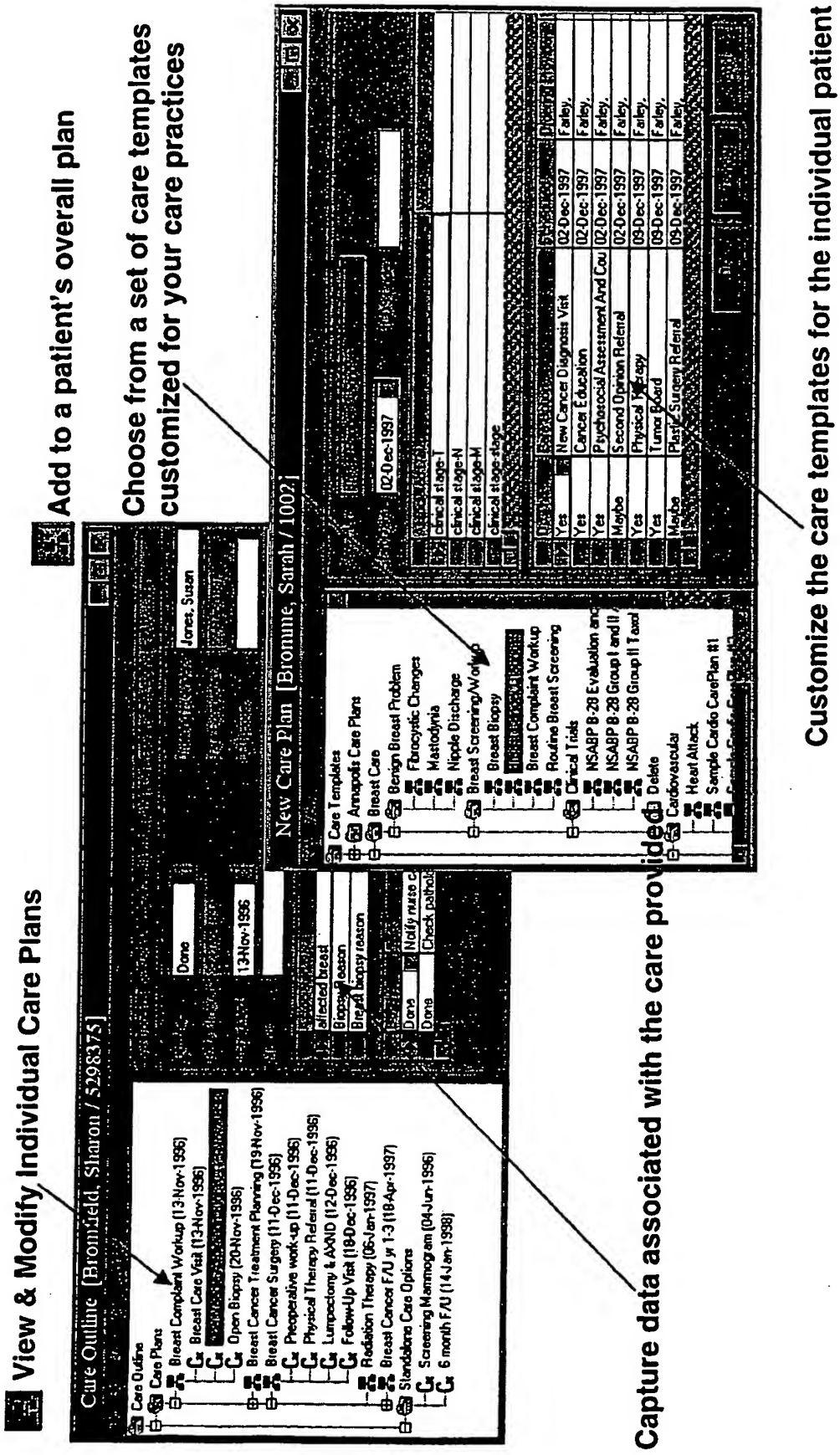

View & Modify Individual Care Plans
Add to a patient's overall plan
Choose from a set of care templates customized for your care practices
Capture data associated with the care provided
Customize the care templates for the individual patient
Care Outline [Bromfield, Sharon / 5298375]
New Care Plan [Bromme, Sarah / 1002]

FIG. 9

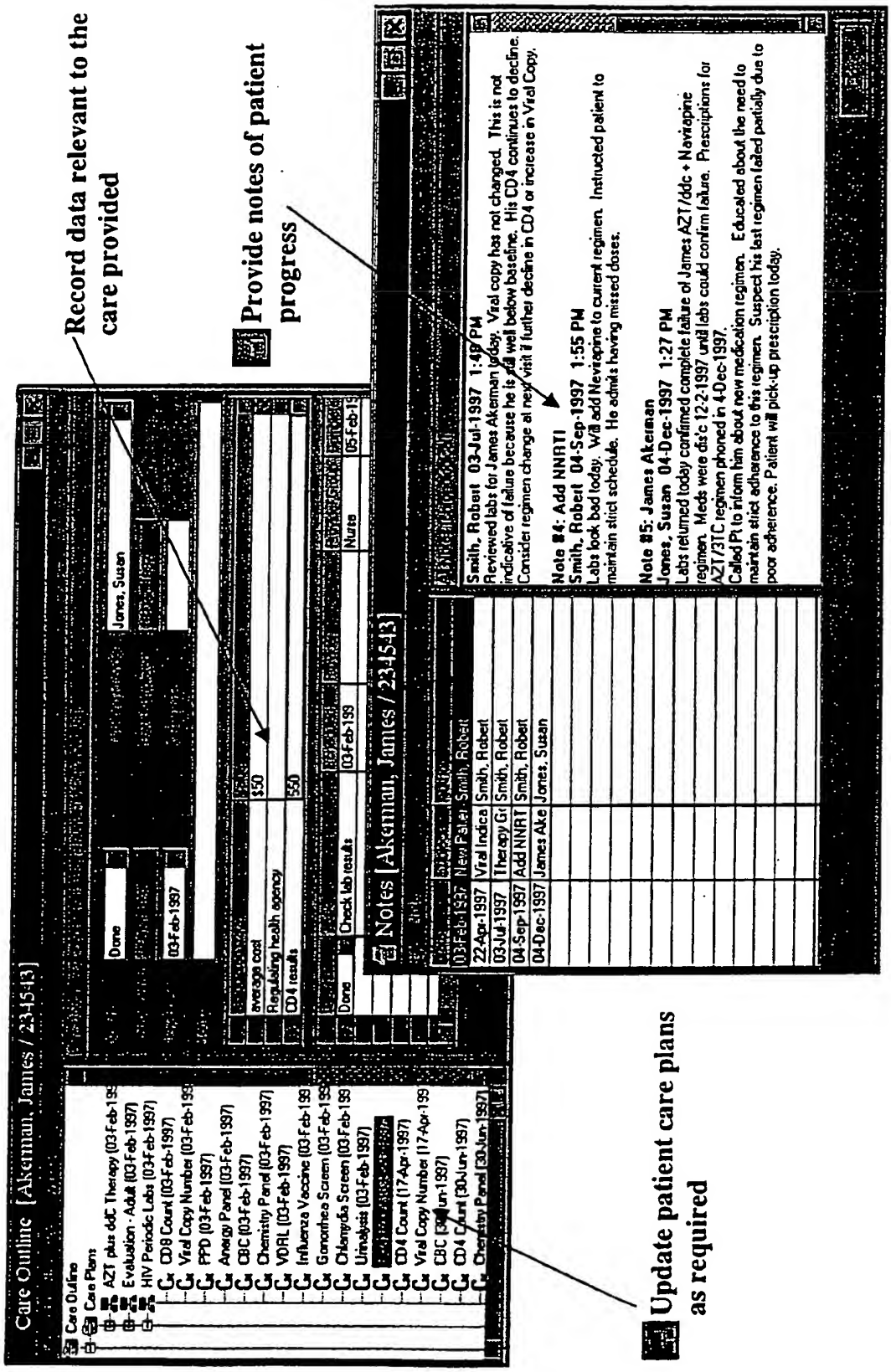

Record data relevant to the care provided
Provide notes of patient progress
Update patient care plans as required
Care Outline [Akerman, James / 234543]
Notes [Akerman, James / 234543]
Jones, Susan
Smith, Robert 03-Jul-1997 1:49 PM
Reviewed labs for James Akerman today. Viral copy has not changed. This is not indicative of failure because he is still well below baseline. His CD4 continues to decline. Consider regimen change at next visit if further decline in CD4 or increase in Viral Copy.
Note #4: Add NNRTI
Smith, Robert 04-Sep-1997 1:55 PM
Labs look bad today. Will add Nevirapine to current regimen. Instructed patient to maintain strict schedule. He admits having missed doses.
Note #5: James Akerman
Jones, Susan 04-Dec-1997 1:27 PM
Labs returned today confirmed complete failure of James AZT/ddc + Nevirapine regimen. Meds were d/c 12-2-1997 until labs could confirm failure. Prescriptions for AZT/3TC regimen phoned in 4-Dec-1997.
Called Pt to inform him about new medication regimen. Educated about the need to maintain strict adherence to this regimen. Suspect his last regimen failed partially due to poor adherence. Patient will pick-up prescription today.

FIG. 10

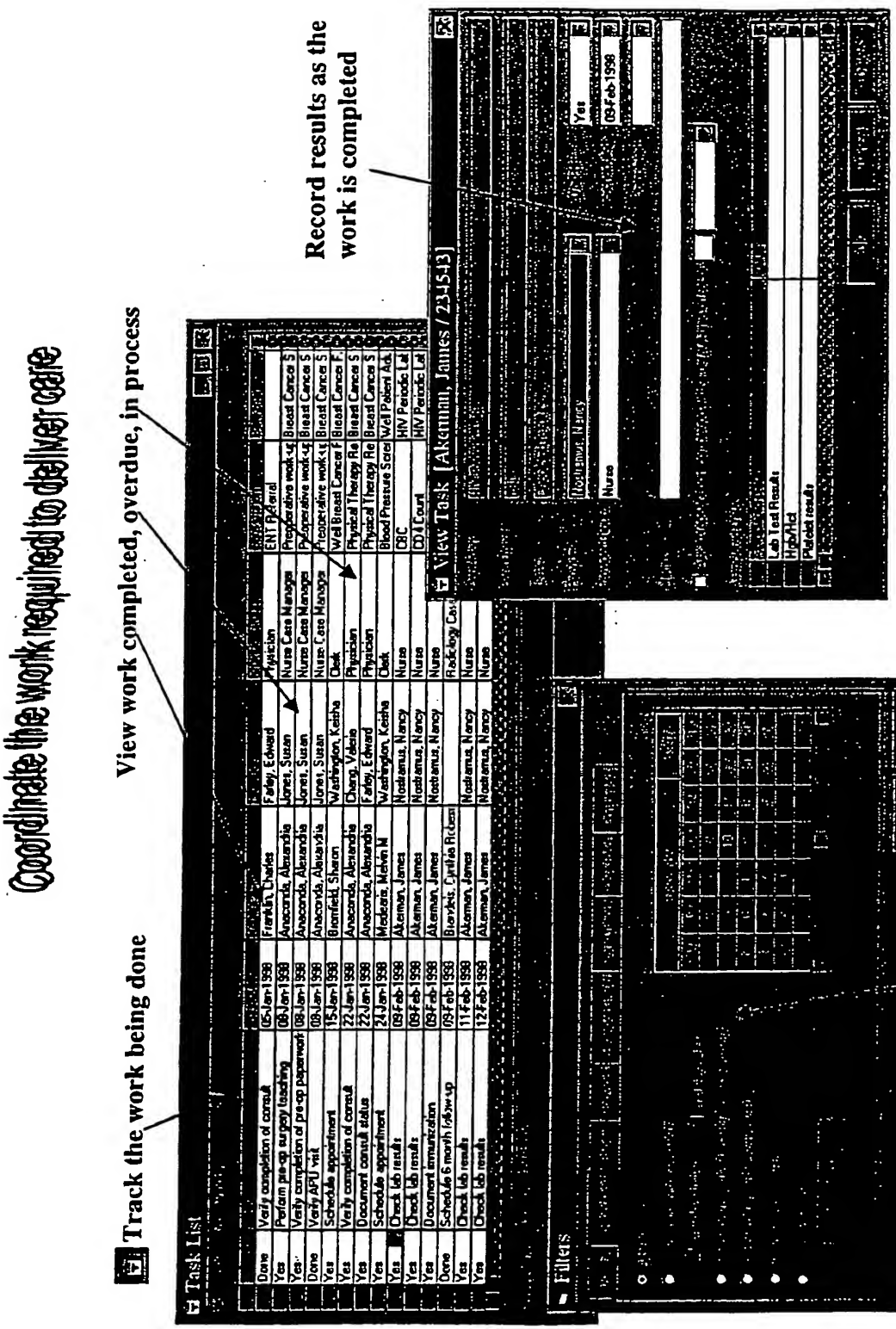

Coordinate the work required to deliver care
Track the work being done
View work completed, overdue, in process
Record results as the work is completed
Filter the tasks list to most efficiently view the work being done
Task List
Filters
View Task [Akeman, James / 234543]

FIG. 11

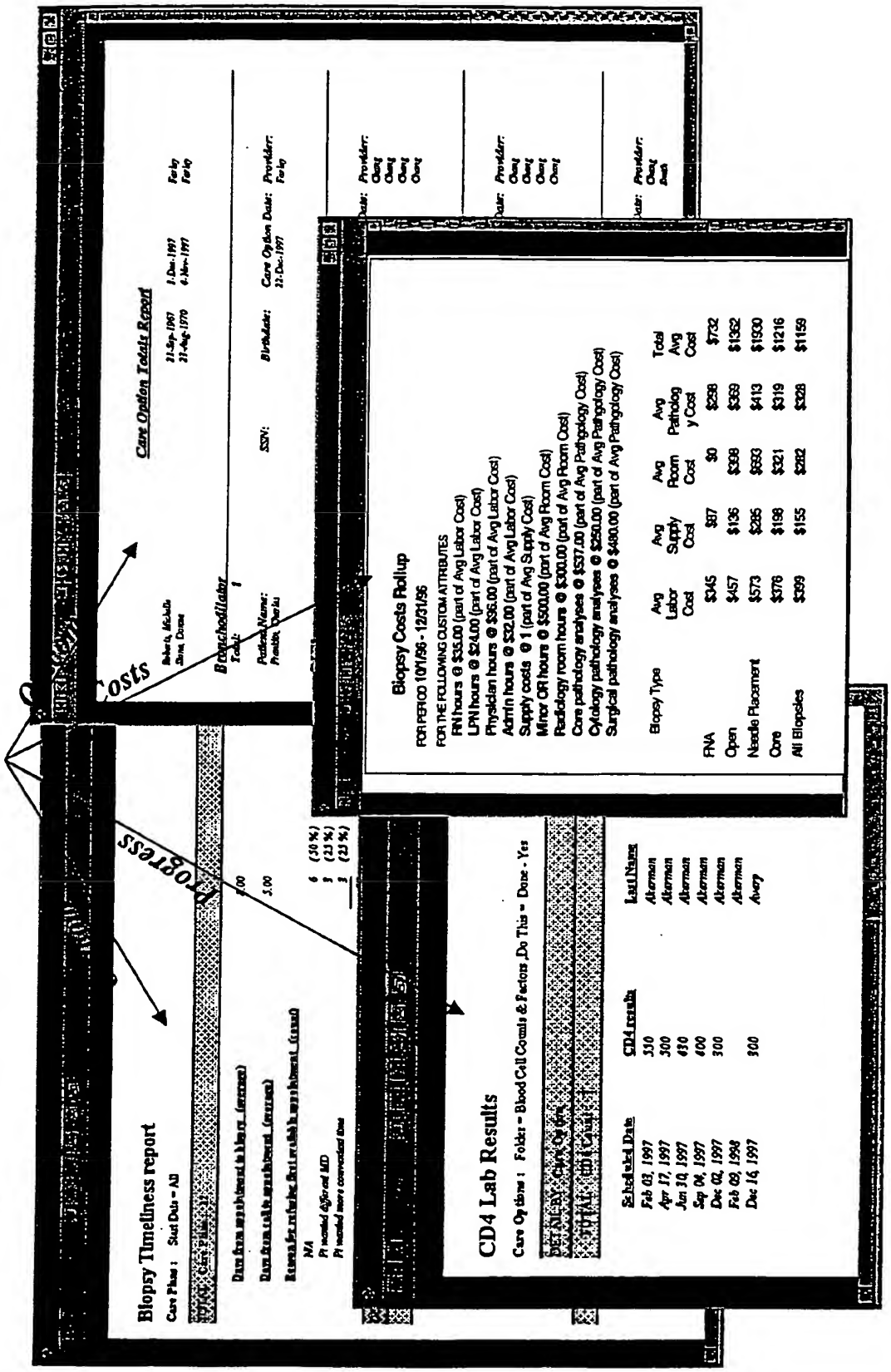

Design & generate reports on the clinical data collected
Biopsy Timeliness report
Care Plan : Start Date - All
Costs
Care Option Totals Report
Biopsy Costs Rollup
FOR PERIOD 10/1/96 - 12/31/96
FOR THE FOLLOWING CUSTOM ATTRIBUTES
RN hours @ $35.00 (part of Avg Labor Cost)
LPN hours @ $24.00 (part of Avg Labor Cost)
Physician hours @ $56.00 (part of Avg Labor Cost)
Admin hours @ $32.00 (part of Avg Labor Cost)
Supply costs @ 1 (part of Avg Supply Cost)
Minor OR hours @ $500.00 (part of Avg Room Cost)
Radiology room hours @ $300.00 (part of Avg Room Cost)
Core pathology analyses @ $537.00 (part of Avg Pathology Cost)
Cytology pathology analyses @ $250.00 (part of Avg Pathology Cost)
Surgical pathology analyses @ $480.00 (part of Avg Pathology Cost)
Biopsy Type
Avg Labor Cost
Avg Supply Cost
Avg Room Cost
Avg Patholog y Cost
Total Avg Cost
FNA $345 $87 $0 $298 $732
Open $457 $136 $398 $369 $1362
Needle Placement $573 $285 $893 $413 $1930
Core $376 $198 $321 $319 $1216
All Biopsies $399 $155 $282 $328 $1159
CD4 Lab Results
Care Options : Folder - Blood Cell Counts & Factors ,Do This - Done - Yes
Scheduled Date
CD4 result
Last Name
Feb 03, 1997 550 Akerman
Apr 17, 1997 300 Akerman
Jun 30, 1997 410 Akerman
Sep 04, 1997 600 Akerman
Dec 02, 1997 300 Akerman
Feb 09, 1998 Akerman
Dec 14, 1997 900 Avery

FIG. 12

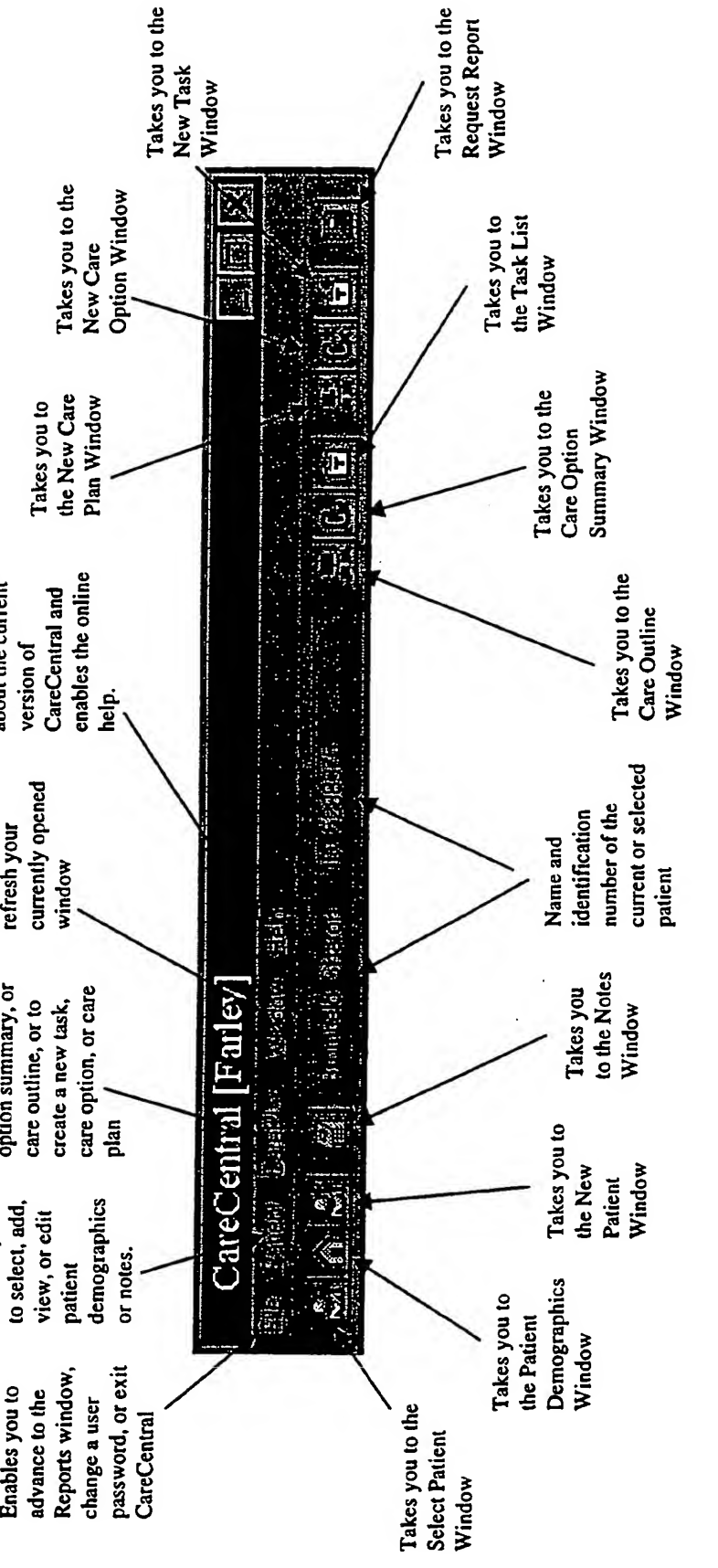

CareCentral [Farley]
Enables you to advance to the Reports window, change a user password, or exit CareCentral
Enables you to select, add, view, or edit patient demographics or notes.
Enables you to view a task list, care option summary, or care outline, or to create a new task, care option, or care plan
Enables you to specify the arrangement or refresh your currently opened window
Provides information about the current version of CareCentral and enables the online help.
Takes you to the Select Patient Window
Takes you to the Patient Demographics Window
Takes you to the New Patient Window
Takes you to the Notes Window
Name and identification number of the current or selected patient
Takes you to the Care Outline Window
Takes you to the Care Option Summary Window
Takes you to the New Care Plan Window
Takes you to the Task List Window
Takes you to the New Care Option Window
Takes you to the Request Report Window
Takes you to the New Task Window

FIG. 13

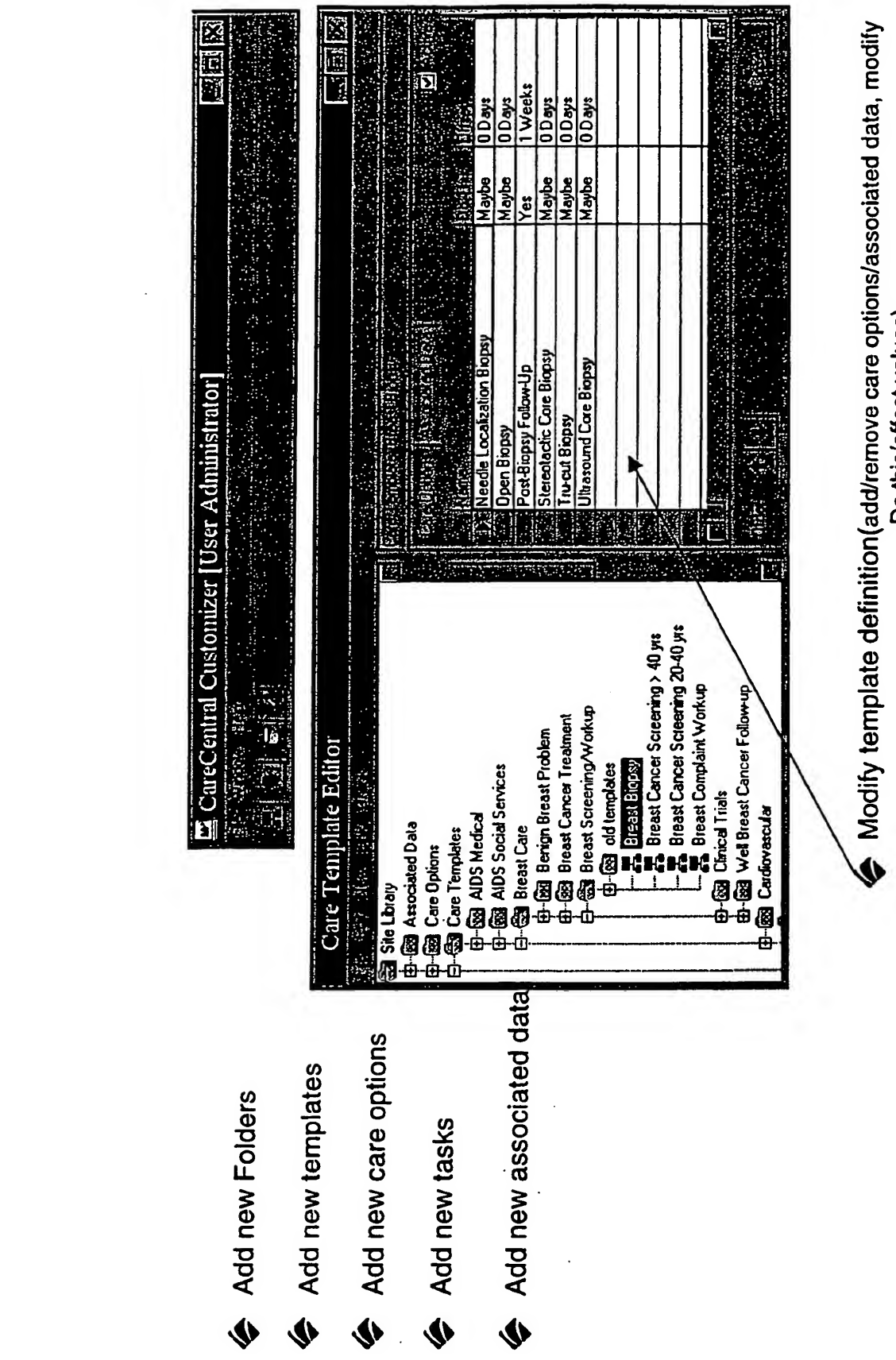

Add new Folders
Add new templates
Add new care options
Add new tasks
Add new associated data
CareCentral Customizer [User Administrator]
Care Template Editor
Site Library
Associated Data
Care Options
Care Templates
AIDS Medical
AIDS Social Services
Breast Care
Benign Breast Problem
Breast Cancer Treatment
Breast Screening/Workup
old templates
Breast Biopsy
Breast Cancer Screening > 40 yrs
Breast Cancer Screening 20-40 yrs
Breast Complaint Workup
Clinical Trials
Well Breast Cancer Follow-up
Cardiovascular
Needle Localization Biopsy Maybe 0 Days
Open Biopsy Maybe 0 Days
Post-Biopsy Follow-Up Yes 1 Weeks
Stereotactic Core Biopsy Maybe 0 Days
Tru-cut Biopsy Maybe 0 Days
Ultrasound Core Biopsy Maybe 0 Days
Modify template definition(add/remove care options/associated data, modify
Do this/offset values)

FIG. 14

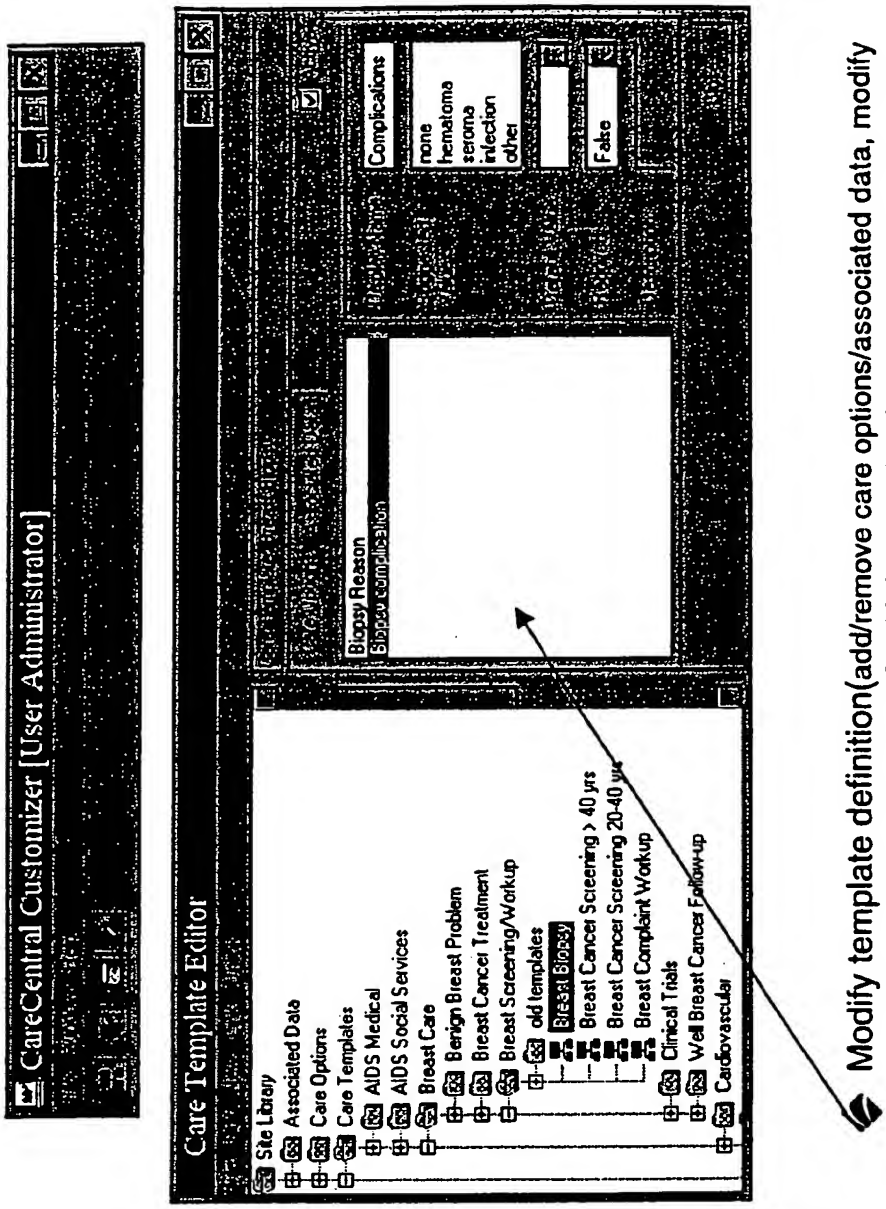

CareCentral Customizer [User Administrator]
Care Template Editor
Site Library
Associated Data
Care Options
Care Templates
AIDS Medical
AIDS Social Services
Breast Care
Benign Breast Problem
Breast Cancer Treatment
Breast Screening/Workup
old templates
Breast Biopsy
Breast Cancer Screening > 40 yrs
Breast Cancer Screening 20-40 yrs
Breast Complaint Workup
Clinical Trials
Well Breast Cancer Follow-up
Cardiovascular
Biopsy Reason
Biopsy complication
Complications
none
hematoma
seroma
infection
other
Fake
Modify template definition(add/remove care options/associated data, modify
Do this/offset values)

FIG. 15

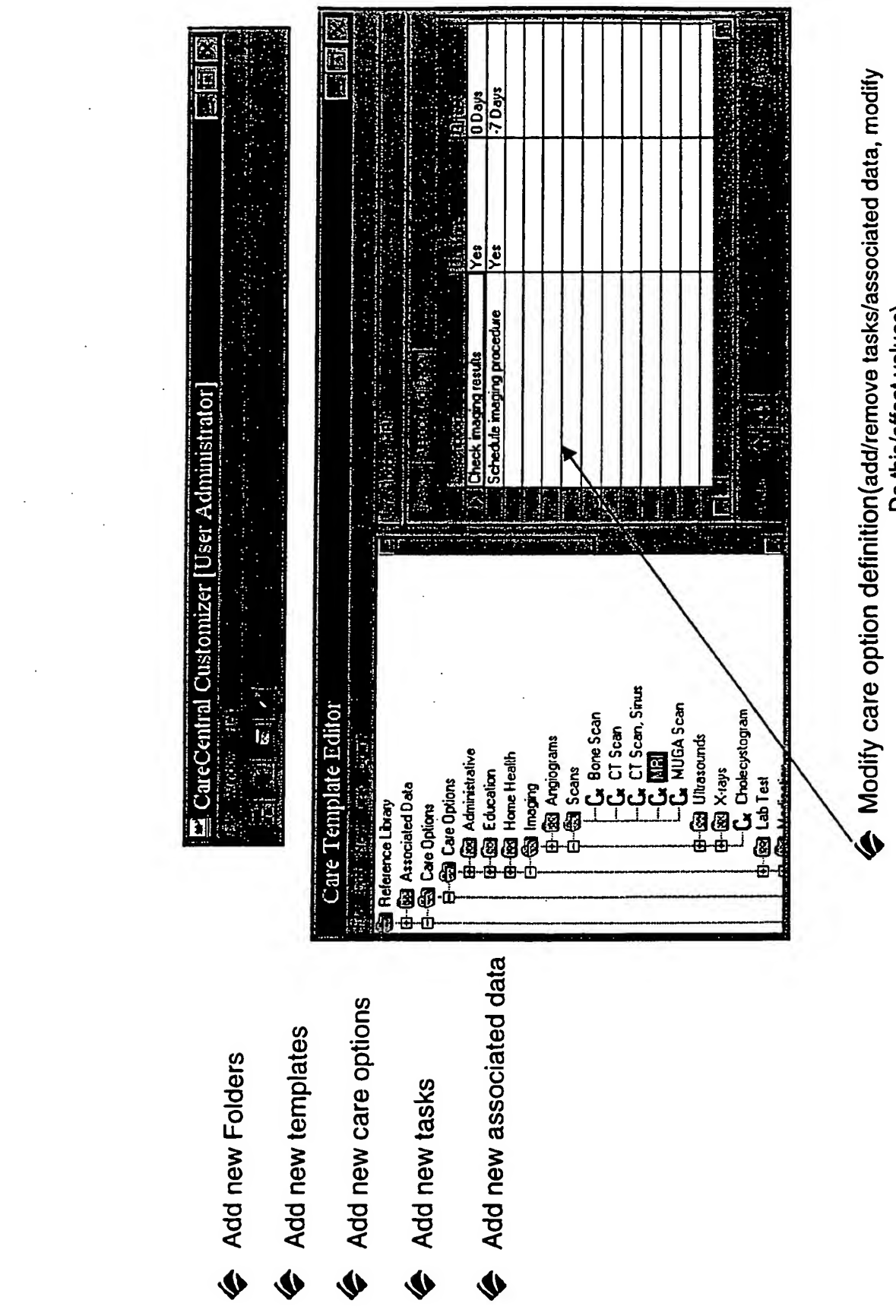

Add new Folders
Add new templates
Add new care options
Add new tasks
Add new associated data
CareCentral Customizer [User Administrator]
Care Template Editor
Reference Library
Associated Data
Care Options
Care Options
Administrative
Education
Home Health
Imaging
Angiograms
Scans
Bone Scan
CT Scan
CT Scan, Sinus
MRI
MUGA Scan
Ultrasounds
X-rays
Cholecystogram
Lab Test
Check imaging results
Schedule imaging procedure
Yes
Yes
0 Days
-7 Days
Modify care option definition(add/remove tasks/associated data, modify
Do this/offset values)

FIG. 16

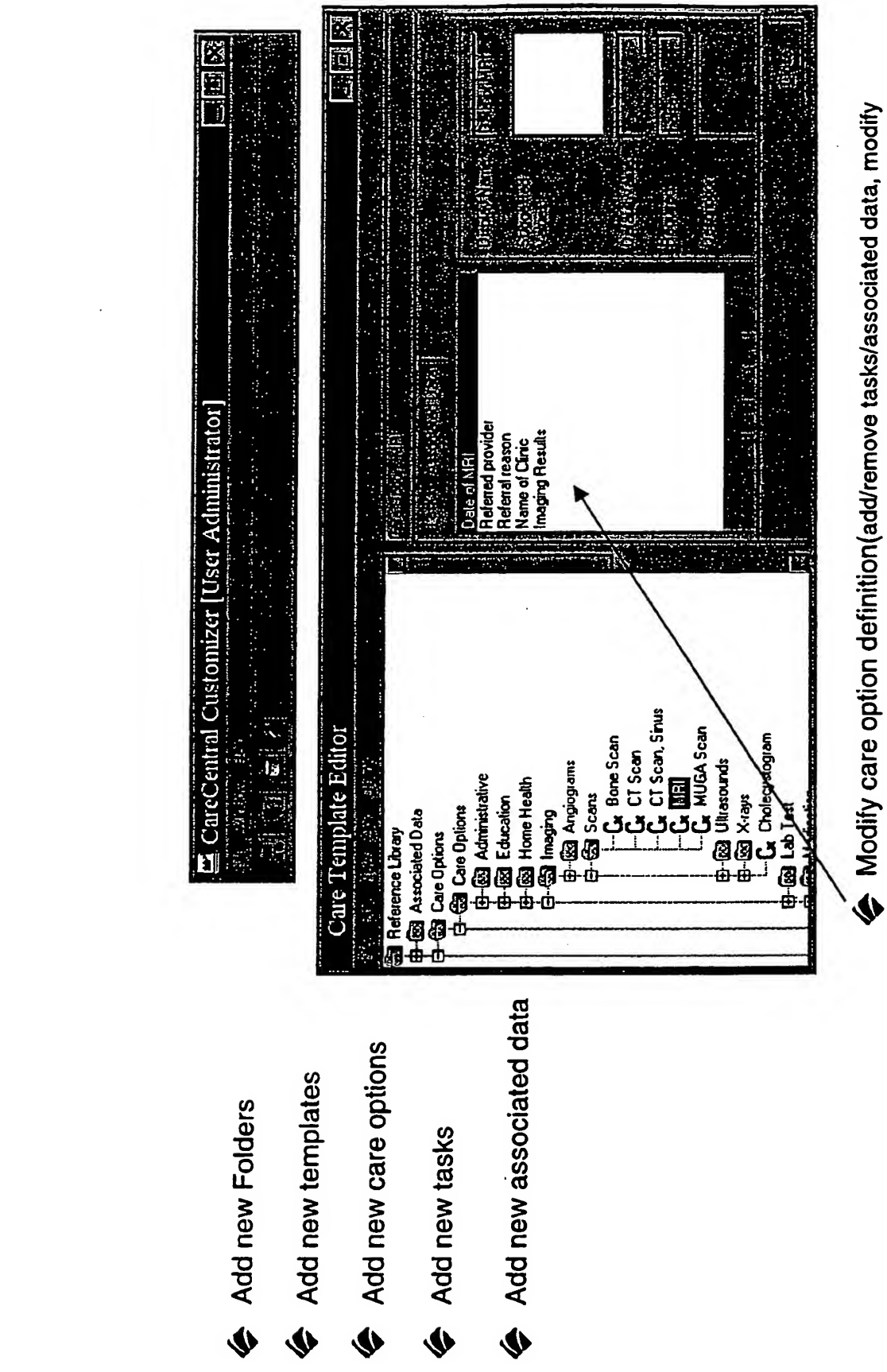

Add new Folders
Add new templates
Add new care options
Add new tasks
Add new associated data
CareCentral Customizer [User Administrator]
Care Template Editor
Reference Library
Associated Data
Care Options
Care Options
Administrative
Education
Home Health
Imaging
Angiograms
Scans
Bone Scan
CT Scan
CT Scan, Sinus
MRI
MUGA Scan
Ultrasounds
X-rays
Cholecystogram
Lab Test
Date of MRI
Referred provider
Referral reason
Name of Clinic
Imaging Results
Modify care option definition(add/remove tasks/associated data, modify
Do this/offset values)

FIG. 17

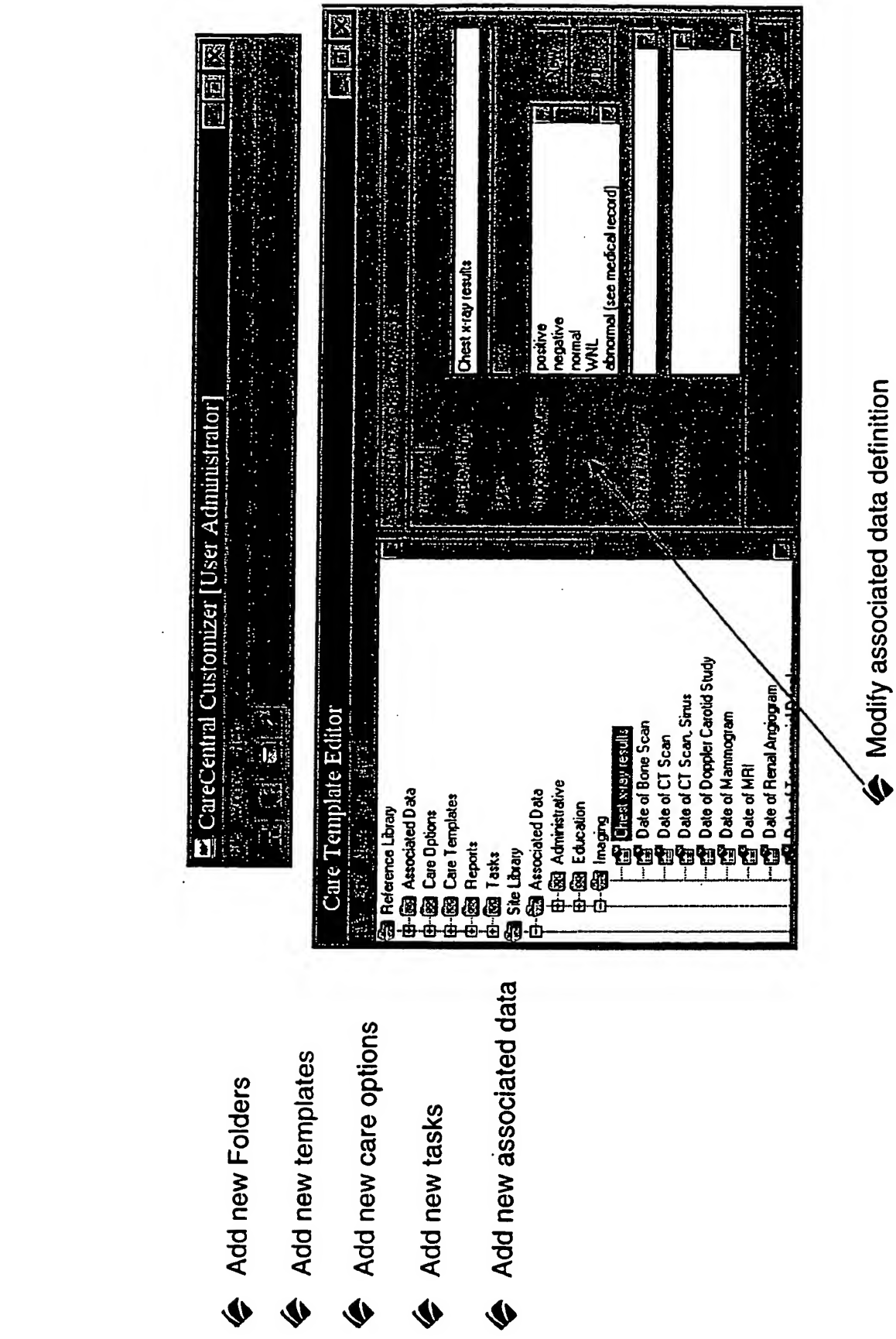

Add new Folders
Add new templates
Add new care options
Add new tasks
Add new associated data
CareCentral Customizer [User Administrator]
Care Template Editor
Reference Library
Associated Data
Care Options
Care Templates
Reports
Tasks
Site Library
Associated Data
Administrative
Education
Imaging
Chest xray results
Date of Bone Scan
Date of CT Scan
Date of CT Scan_Sinus
Date of Doppler Carotid Study
Date of Mammogram
Date of MRI
Date of Renal Angiogram
Chest xray results
positive
negative
normal
WNL
abnormal (see medical record)
Modify associated data definition

FIG. 18

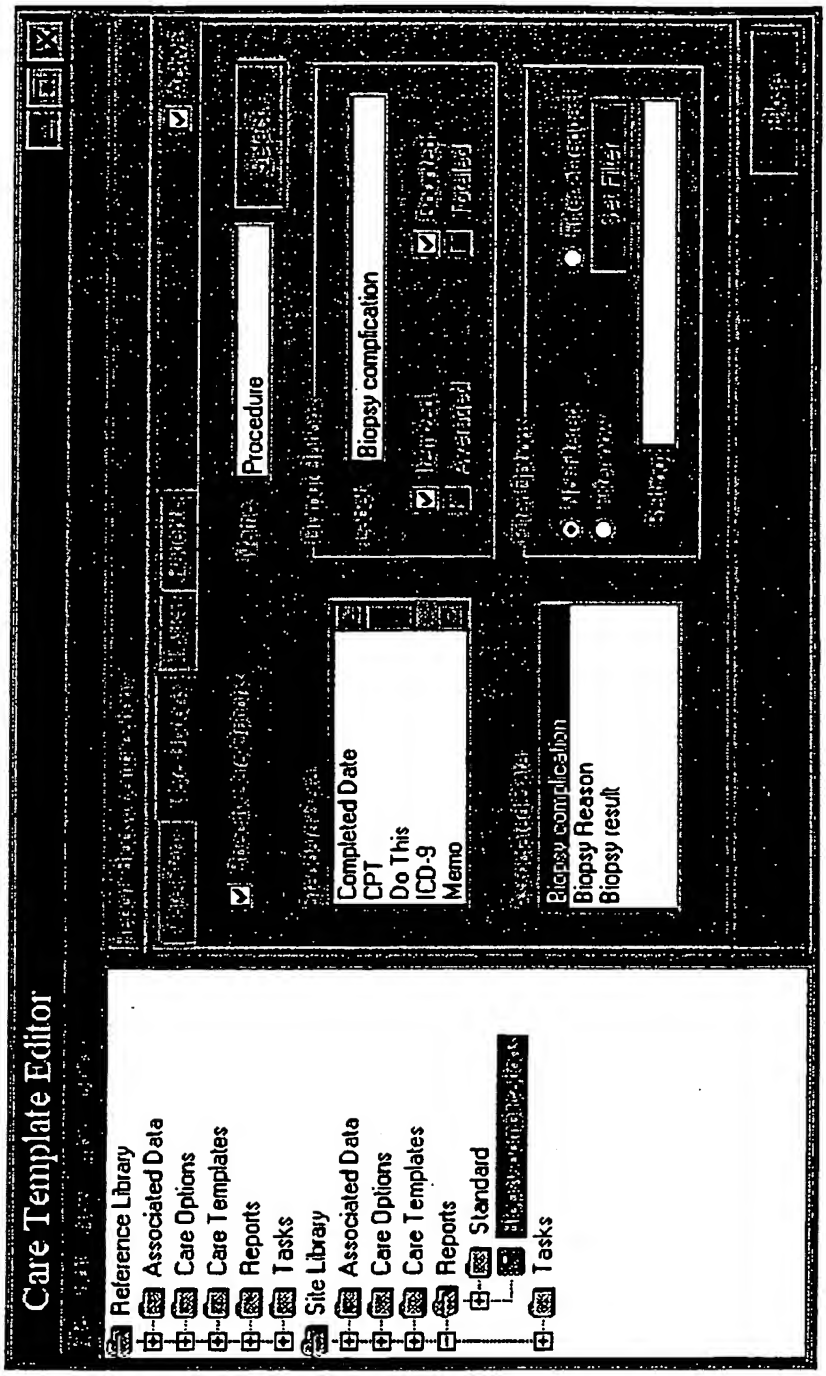

Care Template Editor
Reference Library
Associated Data
Care Options
Care Templates
Reports
Tasks
Site Library
Associated Data
Care Options
Care Templates
Reports
Standard
Tasks
Procedure
Biopsy complication
Completed Date
CPT
Do This
ICD-9
Memo
Biopsy complication
Biopsy Reason
Biopsy result
Set Filter

FIG. 19

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/16973

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) :Please See Extra Sheet.
US CL :604/65; 395/209; 128/897, 732; 705/2, 3
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 604/65; 395/209; 128/897, 732; 705/2, 3

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

NONE

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WEST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 4,464,172 A (LICHTENSTEIN) 07 August 1984, the entire paper is relevant	1-3, and 10
Y	US 5,692,125 A (SCHLOSS et al.) 25 November 1997, the entire paper is relevant	1-3, and 10
Y	US 5,704,371 A (SHEPARD) 06 January 1998, the entire paper is relevant	1-3, and 10
Y	US 5,722,418 A (BRO) 03 March 1998, the entire paper is relevant	1-12
A	US 5,835,897 A (DANG) 10 November 1998, the entire paper is relevant	1-3, 10

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search
05 AUGUST 2000

Date of mailing of the international search report
24 AUG 2000

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer
THUY PARDO James R. Matthews
Telephone No. (703) 305-9707

Form PCT/ISA/210 (second sheet) (July 1998)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/16973

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,913,197 A (KAMEDA) 15 June 1999, the entire paper is relevant	1-12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/16973

A. CLASSIFICATION OF SUBJECT MATTER:
IPC (7):

A61M 1/03; A61B 5/021; G06F 17/60, 15/42, 159/00